

AMENDMENTS TO THE DRAWINGS:

The attached replacement sheets include changes to FIGS. 2, 7, 9, 11-13, 18 and 22. For the convenience of the Examiner, annotated sheets showing the changes is attached. Approval of these changes to the Drawings is respectfully requested.



REPLACEMENT SHEET

Fig. 2

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline ()
{
    //initialize variable
    int c,nl;
    nl=0;
    // count number of lines till EOF is detected
    while ( (c=getchar ()) !=EOF)
        if (c=='\n')
            ++nl;
    // display line count on screen
    printf ("%d\n", nl);
    return nl;
}
```



REPLACEMENT SHEET

Fig. 7

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline ()
{
    //initialize variable
    int c, nl;
    nl=0;
    //count number of lines till EOF is detected
    while ((c=getchar()) !=EOF)
        // check whether it is line feed signal or not
        if (c=='\n')
            // count number of lines in the case of line feed signal
            ++nl;
    // display line count on screen
    printf ("%d\n", nl);
    // return line count to caller
    return nl;
}
```



REPLACEMENT SHEET

Fig. 9

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    int c, nl;
    nl=0;
    while ( (c=getchar()) !=EOF)
        if (c=='\n')
            ++nl;
    printf ("%d\n", nl);
    return nl;
}
```

REPLACEMENT SHEET

Fig. 11

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    //1 initialize variable
    int c, nl;
    nl=0;
    //2 count number of lines till EOF is detected
    while((c=getchar())!=EOF)
        //2.1 check whether it is line feed signal or not
        if (c=='\n')
            //2.1.1 count number of lines in the case of line feed signal
            ++nl;
    //3 display line count on screen
    printf ("%d\n", nl);
    //4 return line count to caller
    return nl;
}
```



REPLACEMENT SHEET

Fig. 12

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    // 1
    int c, nl;
    nl=0;
    // 2
    while ((c=getchar()) != EOF)
        // 2.1
        if (c=='\n')
            // 2.1.1
            ++nl;
    // 3
    printf ("%d\n", nl);
    // 4
    return nl;
}
```



REPLACEMENT SHEET

Fig. 13

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    //1 initialize variable
    int c, nl;
    nl=0;
    //2 count number of lines till EOF is detected
    while ((c=getchar()) !=EOF)
        //2.1
        if (c=='\n')
            //2.1.1
            ++nl;
    //3 display line count on screen
    printf ("%d\n", nl);
    //4
    return nl;
}
```



REPLACEMENT SHEET

Fig. 18

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    //1) initialize variable
    int c, nl;
    nl=0;
    //2) count number of lines till EOF is detected
    while ((c=getchar()) != EOF)
        //2.1) check whether it is line feed signal or not
        if (c=='\n')
            //2.1.1) count number of lines in the case of line feed signal
            ++nl;
    //3) display line count on screen
    printf ("%d\n", nl);
    //4) return line count to caller
    return nl;
}
```




REPLACEMENT SHEET

Fig. 22

```
//outline
// obtain line count of file
// return value
// line count of file
// explanation of parameter
// nothing
int getline()
{
    //2) initialize variable
    int c, nl;
    nl=0;
    //3) count number of lines till EOF is detected
    while ((c=getchar()) != EOF)
        if (c=='\n')
            ++nl;
    //6) display line count on screen
    printf ("%d\n", nl);
    return nl;
}
```